

Test 1: Compsci 101

Kristin Stephens-Martinez

February 14, 2019

Name: _____ (1/2 pt)

NetID/Login: _____ (1/2 pt)

Section Number: _____

Honor code acknowledgment (signature) _____ (1 pt)

	Value	Grade
Front Page	2 pts.	
Problem 1	28 pts.	
Problem 2	6 pts.	
Problem 3	9 pts.	
Problem 4	6 pts.	
Problem 5	5 pts.	
Problem 6	32 pts.	
TOTAL:	88 pts.	

This test has 14 pages be sure your test has them all. Do NOT spend too much time on one question — remember that this class lasts 75 minutes.

In writing code you do not need to worry about specifying the proper `import statements`. Don't worry about getting function or method names exactly right. Assume that all libraries and packages we've discussed are imported in any code you write.

Be sure your name and NetID are legible on this page and that your NetID appears at the top of every page.

There are two blank pages at the end of the test for extra work space.

PROBLEM 1 : (*Name/Type/Value (28 points)*)

Consider the following variables and their values in answering the questions below.

```
words = ['cream', 'teal', 'eggplant', 'sky blue', 'pumpkin', 'peach', 'brown']
say = 'If you are happy and you know it, clap your hands'
```

Each variable in the left column is assigned a value. Provide the type and value of the variable after the assignment. The first two lines have been filled in. Types you can use are **int**, **float**, **list**, **string**, **boolean**.

Assignment	Type	Value
a = 5	int	5
b = words[2]	str	'eggplant'
c = 'H' in say		
d = 'blue' in words		
e = words[6:7]		
f = 5 % 2		
g = len(say.split('h'))		
h = 35 / 10		
i = words[2][3]		
j = words[-1]		
k = 6 // 4		
l = 'you' in say		
m = (4 > 3) and (5 != 6)		
n = 10.0 + 3.0		
o = 'cream' in words		
p = say[:2]		

PROBLEM 2 : (What will Python display? (6 points))

Given the list `food` and the code below, write the output for each `print` statement. **Write the output in the right-column under OUTPUT.**

```
food = ['cucumber', 'beetroot', 'split peas', 'wasabi', 'scallion', 'spinach',  
        'alfalfa sprouts', 'rhubarb', 'collard greens', 'garbanzos beans']
```

```
def mystery(lst, letter):  
    ret = ''  
    for word in lst:  
        if ret.count(letter) < word.count(letter):  
            ret = word  
    return ret
```

CODE	OUTPUT
<pre>print(mystery(food, 'e'))</pre>	
<pre>print(mystery(food, 'a'))</pre>	

PROBLEM 3 : (Spot the Bug (9 points))

Below is a function to calculate the total price of two items, hats and scarves, that have a special discount sale. The regular price for a hat is \$5 and a scarf is \$10. However, a hat and a scarf together are only \$13. After as many hats and scarves are paired up, all the leftover hats or scarves are paid for using the full price. But this function is buggy! Below are some example calls of what the function should return:

Function call	Return value
<code>discount(1, 1)</code>	13 # 1 pair, none left
<code>discount(2, 1)</code>	18 # 1 pair, 1 hat left
<code>discount(2, 3)</code>	36 # 2 pairs, 1 scarf left

```
def discount(hats, scarves):
    if hats > scarves:
        return scarves*13 + (hats-scarves)*5
    if scarves > hats:
        return hats*13 + (scarves-hats)*10
```

Part A (2 points)

In the first cell below, write a *function call* for `discount` with either your own arguments or arguments from the examples that returns a *correct* value. In the second cell, write your function call's return value.

call	return value

Part B (3 points)

In the first cell below, write a *function call* for `discount` with either your own arguments or arguments from the examples that returns a *wrong* value. In the second cell, write your function call's return value. In the third cell, write the value it should return.

call	return value	correct value

Part C (4 points)

Here is the buggy code again. Fix it so that it always returns the correct values.

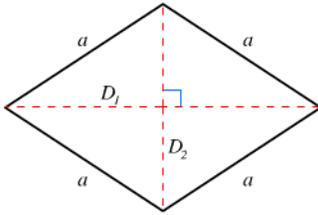
```
def discount(hats, scarves):

    if hats > scarves:

        return scarves*13 + (hats-scarves)*5

    if scarves > hats:

        return hats*13 + (scarves-hats)*10
```

PROBLEM 4 : (Formulas (6 points))**Part A (3 points)**

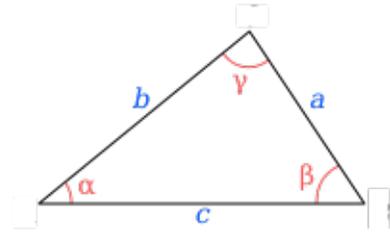
A rhombus is a parallelogram with four equal sides. See the diagram.

The formula to calculate the perimeter of a rhombus in terms of only its diagonals (D_1 and D_2 in the diagram) is:

$$2 * \sqrt{D_1^2 + D_2^2}$$

Implement a function that returns a rhombus's perimeter. Assume that the math module has been imported and you must use the `math.sqrt` function to calculate square roots (e.g. `math.sqrt(4)` evaluates to 2).

```
def rhombusPerimeter(d1, d2):  
    '''  
    d1 (int or float) - length of one diagonal of a rhombus  
    d2 (int or float) - length of the other diagonal of a rhombus  
  
    Return the perimeter of the rhombus as defined by d1 and d2.  
    '''
```

Part B (3 points)

The area of a triangle when only the lengths of all three sides (a , b , and c in the diagram) are known using Heron's formula is defined as:

$$\sqrt{s(s-a)(s-b)(s-c)}$$
$$s = \frac{a+b+c}{2}$$

Implement a function that returns the area of a triangle given only the length of its sides. Assume that the `math` module has been imported and you must use the `math.sqrt` function to calculate square roots (e.g. `math.sqrt(4)` evaluates to 2).

```
def triangleArea(a, b, c):  
    '''  
    a (int or float) - length of the first side of a triangle  
    b (int or float) - length of the second side of a triangle  
    c (int or float) - length of the third side of a triangle  
  
    Return the area of the triangle as defined by a, b, and c using  
    Heron's formula  
    '''
```

PROBLEM 5 : (List Filtering (5 points))

Implement the function `createFilteredList(lst, filter)` as described below. Assuming the following functions are also in the code:

```
def isOdd(n):
    """
    n (int) - Any integer

    Return true if N is odd and false otherwise
    """
    return n % 2 == 1

def divisibleBy3(n):
    """
    n (int) - Any integer

    Return true if N is divisible by 3 and false otherwise
    """
    return n % 3 == 0
```

Example calls are:

Function call	Return value
<code>createFilteredList([1, 2, 3, 4, 5, 6], isOdd)</code>	<code>[1, 3, 5]</code>
<code>createFilteredList([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], divisibleBy3)</code>	<code>[3, 6, 9, 12]</code>

```
def createFilteredList(lst, filter):
    """
    lst (list) - a list of elements that work with the parameter filter
    filter (function) - a function that takes one parameter and will
        work for all the elements in the parameter lst

    Return a new list with only the elements in LST that when called
    with FILTER returns True. The returned list's elements should be in
    the same order as the elements in the original list.
    """
```

PROBLEM 6 : (Toddler Speak (32 points))

Toddlers are well known for mispronouncing words, such as dropping h's and l's or replace p's with t's. In this problem you will write three functions each applying a different rule that converts a word from adult speak to toddler speak. The final function will then convert a normal sentence to how a toddler would say it. *You may assume that all strings are only lower case.*

Part A: Dropping beginning h's (5 points)

Implement the function `startsWithH(word)` as described below. Example calls are:

Function call	Return value
<code>startsWithH('help')</code>	<code>'elp'</code>
<code>startsWithH('shoe')</code>	<code>'shoe'</code>
<code>startsWithH('bath')</code>	<code>'bath'</code>
<code>startsWithH('h')</code>	<code>''</code>

```
def startsWithH(word):  
    '''  
    word (str) - a single word inside of a string  
  
    If WORD starts with the letter 'h', returns WORD without that  
    first letter. If it does not start with the letter 'h', just  
    returns WORD.  
    '''
```

Part B: Dropper inner l's (10 points)

Implement the function `hasAnL(word)` as described below. Example calls are:

Function call	Return value
<code>hasAnL('please')</code>	<code>'pease'</code>
<code>hasAnL('tail')</code>	<code>'tail'</code>
<code>hasAnL('land')</code>	<code>'land'</code>
<code>hasAnL('molehill')</code>	<code>'moehil'</code>
<code>hasAnL('a')</code>	<code>'a'</code>
<code>hasAnL('lo')</code>	<code>'lo'</code>

```
def hasAnL(word):  
    '''  
    word (str) - a single word inside of a string  
  
    Returns a string with all occurrences of the letter 'l' removed  
    from WORD, except for if 'l' is the first or last letter.  
    '''
```

Part C: Dropping ending p's (7 points)

Implement the function `endsWithP(word)` as described below. Example calls are:

Function call	Return value
<code>endsWithP('sheep')</code>	<code>'sheet'</code>
<code>endsWithP('cops')</code>	<code>'cots'</code>
<code>endsWithP('copse')</code>	<code>'copse'</code>
<code>endsWithP('pat')</code>	<code>'pat'</code>

```
def endsWithP(word):  
    '''  
    word (str) - a single word inside of a string  
  
    If WORD ends with 'p', returns a version of WORD with that 'p'  
    replaced with a 't'. If WORD ends with 'ps', returns a version of  
    WORD with that 'ps' replaced with a 'ts'.  
    '''
```

Part D: Toddler Speak in Sentences (10 points)

Implement the function `toddlerSpeak(sentence)` that takes a string with multiple words separated by whitespace. It transforms this string into a string with the words transformed to how a toddler would say it. However, only one rule should ever be applied to a given word. For each word in this string, call the three previous functions in order (`startsWithH`, `hasAnL`, and `endsWithP`). Once a word has been changed by a function, do not transform the word any further.

You must call the functions from the prior parts. Assume they work as specified, regardless of what you wrote. You will get very little credit if you duplicate the functionality represented by those functions.

To be specific, for each word in the sentence, first call it with `startsWithH`. If the word is modified, do not call it with `hasAnL` nor with `endsWithP`. If the word is not modified, call it with `hasAnL` and so on. Below are some example calls. For the first call “help please”, “help” is modified by `startsWithH`, so `hasAnL` and `endsWithP` are not called on it. While “please” is not modified by `startsWithH` and therefore `hasAnL` is called on it. Since that function modified it, `endsWithP` is not called on it.

Function call	Return value
<code>toddlerSpeak('help please')</code>	<code>'elp pease'</code>
<code>toddlerSpeak('sleep time')</code>	<code>'seep time'</code>
<code>toddlerSpeak('the sheep has a little lamb')</code>	<code>'the sheet as a litte lamb'</code>

```
def toddlerSpeak(sentence):
    """
    sentence (str) - a string with words separated by a single whitespace

    Converts a regular sentence into toddler speak.
    """
```

PROBLEM 7 : (*Extra Credit (1 point)*)

Predict what range your percentage grade will be on the exam. If you are correct, you will earn 1 point of extra credit on this exam, rounding in your favor (e.g. your score is 94.5% and you choose the 90%-94% range). The front page has a table with the number of points for each problem.

- 95% - 100%
- 90% - 94%
- 85% - 89%
- 80% - 84%
- 75% - 79%
- 70% - 74%
- 65% - 69%
- 60% - 64%
- 55% - 59%
- 50% - 54%
- 45% - 49%
- 40% - 44%
- 35% - 39%
- 30% - 34%
- 25% - 29%
- 20% - 24%
- 15% - 19%
- 10% - 14%
- 5% - 9%
- 0% - 4%

extra page

extrapage